For...Next Statements

Description

Specifies the beginning and ending points of a For...Next loop. The number of iterations through the loop is controlled by an index variable assigned by the For statement. Optionally, loop termination can be established by conditions specified in the While or Until statements.

Syntax

```
For index = firstindex To lastindex [Step increment]
    [Statements]
[While Conditions] |
[Until Conditions] |
Next [index]
```

Parameters

For…Next statements have the following parameters.

Parameter	Description
index	Also called the "loop variable" or "index variable" assigned the current iteration value. *index* is set to the value of firstindex, and is incremented by the amount specified by the Step amount on each pass through the loop until the lastindex value is exceeded. Then, the loop terminates, and control is passed to the line of code following the Next statement.
firstindex	An integer assigning the beginning value of *index*.
lastindex	An integer indicating the minimum or maximum value of *index*. The value of index is tested against the value of lastindex at the beginning of each successive loop. If the test does not cause termination, the statements between the For and Next statements are executed. When End is exceeded, the loop terminates. Both Start and End (and Step) are evaluated each time through the loop, which can adversely affect performance. It's better to make values for firstindex and lastindex "resolved" values, not expressions such as X = 1 To Count(@RECORD, @FM) + 1
Step	Identifies amount as the amount to increment or decrement index each time the loop is executed. If Step is omitted, the default increment is 1. The index amount can be either positive or negative.
While	Limits or controls the iteration of the For…Next loop. Conditions can be any valid BASIC+ expression(s). If the conditions are false (0), the For…Next loop terminates and program control passes to the statement immediately following the Next statement.
Until	Limits or controls the iteration of the For…Next loop. If the conditions are true (non-zero), the For…Next loop terminates, and program control passes to the statement immediately following the Next statement.

Loop control

Here are some guidelines to follow when writing For…Next statements.

- Each For can have only one Next statement, although For...Next loops may be nested any number of times. Each Next is assumed to be associated with the closest preceding For that has not been paired with a Next. The variable reference following Next is optional, but recommended.

- The Next statement indicates the ending point of the loop. Control then passes to the For statement, where the value of variable is incremented and the termination test is repeated.

- If the loop does terminate, control is transferred to the statement following the Next statement.

- Do not modify the index value in the For...Next loop.

Caution: Do not jump into a For...Next loop using GoTo or GoSub.

See also

Loop, GoTo

Example

```
* This loop adds the squares of integers from 1 to 50.
Total = 0
For X = 1 To 50
Total += X * X
Next
/* Reads a stored procedure from the SYSPROCS table,
and scans through it until a line containing "CREATED BY:" is found.
Variable L will hold the number of the line it was found on. */
Open "SYSPROCS" To proc_file Then
Read @RECORD From proc_file, "TEST_PROCEDURE" Then
  line_count = Count(@RECORD, #"")
  For L = 1 To line_count
   cur_line = @RECORD<L>
  Until Index(cur_line, "CREATED BY:", 1)
  Next
End Else
  GoSub Read_Error
End
End Else
GoSub Open_Error
End
```